UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,149	01/02/2002	Bob Janssen	DVME-1018US	9408

21302   7590   02/21/2008

KNOBLE, YOSHIDA & DUNLEAVY
EIGHT PENN CENTER
SUITE 1350, 1628 JOHN F KENNEDY BLVD
PHILADELPHIA, PA 19103

EXAMINER
SCUDERI, PHILIP S

ART UNIT	PAPER NUMBER
2153	

MAIL DATE	DELIVERY MODE
02/21/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

MAILED

FEB 2 1 2008

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/040,149
Filing Date: January 02, 2002
Appellant(s): JANSSEN ET AL.

Kevin J. Dunleavy
Reg. No. 32,024
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 1/2/2008 appealing from the Office action mailed

7/3/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,909,545	Frese, II et al.	6-1999
5,613,090	Willems	3-1997

Designing Web-Based User Interfaces, Scott Ambler, 29 June 2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-10, 18, and 19 are rejected under 35 U.S.C. 102(b) as being anticipated by

Frese (U.S. Patent No. 5,909,545).

As to claim 1, Frese teaches a server-based computing system, comprising at least one server

(20) and at least one client computer (16), connected to the server (20) through a network, wherein

the server (20) comprises means for providing the client computer (16) with a user interface (web

page), wherein the client computer (16) comprises an input device for providing input to an

application (22) through the user interface (web page) and a display device for presenting output

from an application (22) through the user interface (web page), wherein the server (20) comprises

means for running the application (22), wherein the client computer (16) comprises means for

locally running at least one further application (18), wherein the system comprises means for

controlling the locally run applications (18) through the user interface (web page) provided by the

server (20). (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

The claims specify that the system is "configured to enable" the server to control the display.

Thus, the system merely needs to be <u>capable of enabling</u> the server to control the display. Whether

the server is actually provided with means for controlling the display is therefore irrelevant. To be capable of enabling the server to control a display at the client the server needs to be: (1) capable of being provided with means for controlling the display of the client; and (2) connected to the client via some communication network. Almost any computer in existence is capable of being provided with software for controlling the display of a client to which it is connected via a network. The server (20) disclosed by Frese is a conventional server with various applications installed thereon and thus is no exception.

Moreover, Frese would anticipate the claim even if it actually required the server to control the display on a screen of the display device of a screen area having contents generated locally on the client computer. Specifically, Frese teaches a server (20) that controls the display on a screen of the display device of a screen area (specifies RDM applet 18's parameters) having contents generated locally (RDM applet 18's display) on the client computer (16) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 2, Frese teaches means for controlling an application (22) running on the server (20) and further applications (18), running locally, through the user interface (web page) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 3, Frese teaches that the user interface (web page) comprises means for initiating a locally run application (18) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 4, Frese further teaches means for presenting an overview of available applications installed on the server and on the client computer through the user interface (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 5, Frese teaches that the user interface comprises means for presenting an overview of applications running on the server (20) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 6, Frese teaches that the user interface (web page) comprises means for generating a merged local client screen for display on the display device (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 7, Frese teaches that the server (20) comprises means for controlling the display of the merged local client screen on the display device (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 8, Frese teaches that the client computer (16) comprises means for generating a local client screen area (browser), comprising visual output from the locally run applications (18), and the server (20) comprises means for generating a screen area, wherein the system comprises means for merging the local client screen area and the screen area generated by the server, to form a local client screen (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 9, Frese teaches means for automatically updating the local client screen, when changes occur in the local client screen or in the screen area generated by the server (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 10, Frese teaches means for selecting a running application; and means for providing input to the selected application or means for presenting output from the selected application on the client computer (16) through the user interface (web page) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 18, Frese teaches a computer program stored on a computer readable medium, wherein the computer program can be loaded onto a server (20) through a network to a client

computer (16) wherein the client computer (16) comprises an input device for providing an input to

an application (22) through a user interface (web page) and a display device for presenting output

from an application (22) through the user interface (web page), wherein the server (20) comprises

means for running the application (22), wherein the client computer (16) comprises means for

locally running at least one further application (18), and wherein the server (20) and client computer

(16) comprise means for controlling the locally run applications (18) through a user interface (web

page) provided by the server (20), wherein the computer program, when run on the server (20)

instructs the server (20) to provide the client computer (16) with the user interface (web page) and

the server (20) controls (by specifying applet parameters) the display on a screen of the display

device having contents generated locally on the client computer (16) (see Frese at fig. 1; col. 6, ll. 39

to col. 8, ll. 50).

As to claim 19, Frese teaches a computer program stored on a computer readable medium,

wherein the computer program can be loaded onto a computer (16), the computer (16) being

connected through a network to a server (20), and comprising an input device for providing input to

an application (22) through a user interface (web page) and a display device for presenting output

from an application (22) through the user interface (web page), wherein the server (20), comprises

means for running an application (22), and wherein the computer comprises means for locally

running at least one further application (18), wherein the computer program, when run on the

computer (16), causes the computer to accept the user interface (web page), the user interface (web

page) being configured for controlling the at least one locally run application (18) and being

provided by the server (20), and further causes the computer (16) to display a screen area having

contents generated locally (RDM applet 18's display) on the client computer (16) according to

display properties (applet parameters) specified by (in an APP tag) the server (20) (see Frese at fig. 1;

col. 6, ll. 39 to col. 8, ll. 50).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness

rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-10, 18, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Willems (U.S. Patent No. 5,613,090).

As to claim 1, in regards to figure 8, Willems describes a prior art X Windows environment,

comprising at least one server (X server) and at least one client computer, connected to the server

(X server) through a network, wherein the client computer comprises an input device for providing

input to an application (X applications) through a user interface (remote window manager 100) and

a display device for presenting output from an application (X applications) through the user

interface (remote window manager 100), wherein the server (X server) comprises means for running

the application (X applications) (see Willems at fig. 8; col. 13, ll. 39-58).

Willems does not expressly disclose that the server (X server 102) comprises means for

providing the client computer with a user interface (remote window manager 100). However,

Willems does not teach away from running the window manager 100 on the same computer as the

X server. Willems discloses there is a performance hit taken when the server (X server 102) and the

means for providing the user interface (remote window manager 100) are on separate machines (see

Willems at col. 13, ll. 53-58). It would have been obvious to one of ordinary skill in the art to install the X server 102 and the remote window manager 100 on the same computer to avoid this performance hit (see id.)

Willems does not expressly disclose that the client computer comprises means for locally running at least one further application. Nonetheless, it was well known in the art to provide a client computer with means for running local applications. It would have been obvious to provide such a means here so that users can run their own local applications.

The prior art shown in figure 8 of Willems does not teach controlling locally run applications through the user interface (window manager 100) provided by the server (X server). However, enabling a window manager (i.e., a user interface) to control locally run applications was well known in the art, as evidenced by Willems discussion in regards to figure 9.

In regards to figure 9, Willems describes a window manager that controls a client's locally run applications and provides advantages such as reducing the amount of front-end code required. See Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13. It would have been obvious to modify the prior art window manager of figure 8 by enabling it to control a client's locally run applications to reduce the amount of front-end code required (see Willems at col. 14, ll. 1-5).

Claim 1 recites "the client computer ... is configured to enable the server ... to control the display on a screen of the display device ... of a screen area having contents generated locally on the client computer" (emphasis added). To meet this limitation Willems' system merely needs to be capable of enabling the server (X server) to control the display on a screen of the display device of a screen area having contents generated locally on the client computer (16) (see MPEP § 2111.04). Willems' client computer is connected to the server (X Server) via a network and therefore the

system <u>capable of enabling</u> the server (X Server) to control the display on a screen of the display device of a screen area having contents generated locally on the client computer (see Willems at fig. 9).

Moreover, Willems would obviate the claims even if they actually required the server to control the display on a screen of the display device of a screen area having contents generated locally on the client computer because enabling the window manager to control a client's locally run applications would enable it to control the display on a screen area of the display device of a screen area having contents generated locally on the client computer (see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13).

As to claim 2, Willems further teaches means for controlling an application running on the server and further applications, running locally, through the user interface (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 3, Willems does not expressly disclose that the user interface comprises means for initiating a locally run application. However, it was well known in the art that X Windows environments provide means for initiating locally run applications. It would have been obvious to provide means for initiating locally run applications here to enable users to conveniently initiate the locally run applications (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claims 4 and 5, Willems does not expressly disclose presenting an overview of available applications installed on the client and/or the server. However, it was well known in the art that X Windows environments provide means for presenting an overview of available applications. It would have been obvious to provide means for presenting an overview of available applications to enable users to conveniently locate available applications.

As to claim 6, Willems further teaches means for generating a merged local client screen, for display on the display device (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13)

As to claim 7, Willems further teaches that the server comprises means for controlling the display of the merged local client screen on the display device (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 8, Willems further teaches that the client computer comprises means for generating a local client screen area, comprising visual output from the locally run applications, and the server comprises means for generating a screen area, wherein the system comprises means for merging the local client screen area and the screen area generated by the server, to form the local client screen area (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 9, Willems further teaches means for automatically updating the local client screen, when changes occur in the local client screen, when changes occur in the local client screen area and/or in the screen area generated by the server (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 10, Willems further teaches means for selecting a running application; and means for providing input to the selected application and/or means for presenting output from the selected application on the client computer through the user interface (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 18, in regards to figure 8, Willems describes a prior art X Windows environment, comprising a computer program stored on a computer readable medium, wherein the computer program can be loaded onto a server (X server) connected through a network to a client computer wherein the client computer comprises an input device for providing input to an application (X

applications) through a user interface (window manager 100), wherein the server (X server) comprises means for running the application (X applications) (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

Willems does not expressly disclose that the server (X server 102) provides the client computer with the user interface (remote window manager 100). However, Willems does not teach away from running the window manager 100 on the same computer as the X server. Willems discloses there is a performance hit taken when the server (X server 102) and the means for providing the user interface (remote window manager 100) are on separate machines (see Willems at col. 13, ll. 53-58). It would have been obvious to one of ordinary skill in the art to install the X server 102 and the remote window manager 100 on the same computer to avoid this performance hit (see id.)

Willems does not expressly disclose that the client computer comprises means for locally running at least one further application. Nonetheless, it was well known in the art to provide a client computer with means for running local applications. It would have been obvious to provide such a means here so that users can run their own local applications.

The prior art shown in figure 8 of Willems does not teach controlling locally run applications through the user interface (window manager 100) provided by the server (X server). However, enabling a window manager (i.e., a user interface) to control locally run applications was well known in the art, as evidenced by Willems discussion in regards to figure 9.

In regards to figure 9, Willems describes a window manager that controls a client's locally run applications and provides advantages such as reducing the amount of front-end code required. (see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13). It would have been obvious to modify the

prior art window manager of figure 8 by enabling it to control a client's locally run applications to reduce the amount of front-end code required (see Willems at col. 14, ll. 1-5).

Willems obviate the limitation "the server controls the display on a screen of the display device of a screen area having contents generated locally on the client computer" because enabling the window manager to control a client's locally run applications would enable it to control the display on a screen area of the display device of a screen area having contents generated locally on the client computer (see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13).

As to claim 19, Willems teaches a computer program stored on a computer readable medium, wherein the computer program can be loaded onto a computer, the computer being connected through a network to a server (X server), and comprising an input device for providing input to an application (X applications) through a user interface (window manager 100) and a display device for presenting output from an application (X applications) through the user interface (window manager 100), wherein the server (X server), comprises means for running an application (X applications). (see Willems at fig. 8; col. 13, ll. 39-58).

Willems does not expressly disclose that the <u>server</u> (X server 102) provides the client computer with the user interface (remote window manager 100). However, Willems does not teach away from running the window manager 100 on the same computer as the X server. Willems discloses there is a performance hit taken when the server (X server 102) and the means for providing the user interface (remote window manager 100) are on separate machines (see Willems at col. 13, ll. 53-58). It would have been obvious to one of ordinary skill in the art to install the X server 102 and the remote window manager 100 on the same computer to avoid this performance hit (see id.)

Willems does not expressly disclose that the computer comprises means for locally running

at least one further application. Nonetheless, it was well known in the art to provide a client

computer with means for running local applications. It would have been obvious to provide such a

means here so that users can run their own local applications.

The prior art shown in figure 8 of Willems does not teach that the user interface (window

manager 100) controls the at least one locally run application and causes the computer to display a

screen area having contents generated locally on the client computer according to display properties

specified by the server. However, enabling a window manager (i.e., a user interface) to control

locally run applications was well known in the art, as evidenced by Willems discussion in regards to

figure 9.

In regards to figure 9, Willems describes a window manager that controls a client's locally

run applications and provides advantages such as reducing the amount of front-end code required.

(see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13). It would have been obvious to modify the

prior art window manager of figure 8 by enabling it to control a client's locally run applications to

reduce the amount of front-end code required. (see Willems at col. 14, ll. 1-5). Enabling the

window manager of figure 8 to control the client's locally run applications would cause the user

interface to display a screen area having contents generated locally on the client computer according

to display properties specified by the server.

(10) Response to Argument

A.   U.S. Patent No. 5,909,545 to Frese, II et al.

      1.   Independent Claim 1 and Dependent Claims 2, 3, 5, and 10

a.   In the Final Rejection the examiner took the position that for the system to be "configured

to enable the server ... to control the display on a screen area having contents generated locally on

the client computer" the server needs to be: (1) capable of being provided with means for

controlling the display of the client; and (2) connected to the client via some communication

network. (Final Rejection mailed 7/3/2007 at pp. 10).

       Applicant argues that the examiner's interpretation was improper. (Brief at pp. 8).

Applicant argues that "merely connecting a client computer 16 to a server 20 is not sufficient to

enable the server to control the display device of a screen area having contents generated locally on

the client computer" (emphasis added). (Brief at pp. 8). Applicant argues that "for the server to be

enabled to control the display device of a screen area having contents generated locally on the client

computer the server must include some hardware or software for controlling the display of the local

client computer" (emphasis added) (Brief at pp. 8, ll. 5-7). Applicant argues that the claims, page

11, lines 3-6 of the specification, and *Boston Sci. Corp. v. Cordis Corp.*, 2006 U.S. Dist. LEXIS 94329

(D. Cal. 2006) ("configured to" was interpreted to mean "intentionally and specifically made to act

in a certain way") make this clear.

       These arguments are not deemed persuasive for the following reasons.

       The claims limit the system (not the server) to being "configured to enable the server ... to

control the display ...." For example, claim 1, lines 7-10 recite "the system ... is configured to

enable the server (1) to control the display on a screen of the display device (7)..." (emphasis

added).

       Applicant cites the decision in *Boston Sci. Corp. v. Cordis Corp.* In view of that decision, the

examiner acknowledges that in view of this decision "configured to" generally means "intentionally

and specifically made to act in a certain way." Here, the system is therefore required to be

intentionally and specifically made to <u>enable</u> the server to control the client's display. Applying the broadest reasonable interpretation, this means that (1) the server and the client need to be connected via some network and (2) the server needs to be capable of being provided with means for controlling the display of the client, as described in the Final Rejection. (See Final Rejection mailed 7/3/2007 at pp. 10). Frese clearly discloses these features and therefore meets the claims. (See, e.g., Frese at fig. 1).

The examiner has reviewed page 11, lines 3-6 of the specification yet maintains the rejection. This section of the specification discloses that "the <u>server</u> comprises means for controlling the display of the local client screen 16 on the screen 7 of the client computer" (emphasis added). As stated by applicant, this portion of the specification suggests that the server has some hardware or software means for controlling the display on the client. But, again, the claims limit the <u>system</u> (<u>not</u> the client or the server) to being "configured to enable the server ... to control the display ...." Indeed, applicant <u>could</u> have limited the claims such that the server requires "means for controlling the display ...", but chose to use the "system ... is configured to enable" language instead.

b.   Applicant argues that the examiner "has not identified any structure on the <u>client computer</u> configured to enable the server to control the display on a screen of the display device" (emphasis added). (Brief at pp. 9, ll. 20-22). Applicant further argues that "it is the client computer of claim 1 that must be configured to enable ... and not the server as the Examiner mistakenly concludes." (Brief at pp. 10, ll. 1-2).

These arguments are not deemed persuasive for similar reasons to those discussed in section (A)(1)(a) above. Again, the claims limit the <u>system</u> (<u>not</u> the client) to being "configured to enable the server ... to control the display ...." For example, claim 1, lines 7-10 recite "the <u>system</u> ... is

configured to enable the server (1) to control the display on a screen of the display device (7)..."

(emphasis added).

Applicant appears to be jumping back and forth between arguing that the server needs to be

"configured to enable ..." and that the client needs to be "configured to enable ...." But, again,

claim 1, lines 7-10 recite "the system ... is configured to enable the server (1) to control the display

on a screen of the display device (7)..." (emphasis added). The system of course comprises at least

all the claimed elements such as the server, the client, and the network. (See, e.g., claim 1).

c.   Applicant argues that "the HTML page described at col. 7, lines 33-35 of Frese, is not an

application that is run locally, as the Examiner suggests." (Brief at pp. 10).

This argument is not deemed persuasive. The examiner agrees that the HTML page is not a

locally run application. But, the examiner never intended to suggest that the HTML page was a

locally run application. Rather, it is the examiner's position that RDM applet 18 corresponds to the

locally run application. For example, page 10 of the Final Rejection states that Frese teaches that

"means for locally running at least one further application (18), wherein the system comprises means

for controlling the locally run applications (18)" (emphasis added). Reference number 18 in Frese

corresponds to the RDM applet, which is run locally on the client. (See, e.g., Frese at fig. 1). The

HTML page corresponds to the claimed "user interface."

d.   In the Final Rejection the examiner took the position that Frese controls some aspects of the

display by specifying parameters for the RDM applet (Final Rejection mailed 7/3/2007 at pp. 11).

Applicant argues that this analysis of Frese is improper (Brief at pp. 11). Namely, Applicant

argues that "[t]he applet tag of the HTML document is used to select which RDM applet is to be

transmitted to the local computer for execution on the local computer and not for the purpose of controlling display parameters." (Brief at pp. 11).

This argument is not deemed persuasive for the following reasons.

The examiner agrees that the applet tags in the HTML document are used to select an RDM applet that best corresponds to the client's operating environment. Frese shows such tags in col. 7, ll. 46-55. And, Frese states that the "operating system environment identification is used to select a RDM which best corresponds to the operating system environment." (Frese at col. 10, ll. 9-11). For example, the client could select the "rdm", "rdm_internet", or "rdm_ether" applet shown in col. 7.

However, this does not mean that the server does not "control" the display by specifying the applets and parameters shown by Frese in col. 7 within the broadest reasonably interpretation of "control." Each of the applet options listed in col. 7 have parameters that are part of the HTML code sent from the server. For example, the client could choose the "rdm" applet with the parameters "platform=win31" and "colordepth=8bit" etc. (See Frese at col. 7). Because the server provides the applet tags and their parameters that the client is able to select, the server ultimately controls the look and feel of the applet that is run by the client.

e.   Applicant argues that the examiner asserts that the HTML document containing the applet tag is stored at the server 20 of Frese. (Brief at pp. 11). Applicant argues that this assertion is not supported by Frese. (id.) Applicant argues that Frese "can present and transmit HTML documents over the internet, but nowhere indicates that these HTML documents are actually stored at server 20 as the Examiner assumes." (id.)

These arguments are not deemed persuasive for the following reasons.

Upon further review of the Frese reference the examiner agrees with applicant that Frese

does not actually state that the HTML documents are stored at the server. However, the server does

not actually need to store the HTML documents in order to meet the claim. Frese's server clearly

provides the HTML documents to the client and therefore "controls" the display as described

above.

Moreover, the examiner is inclined to point out that if the rejection over Frese were ever to

be withdrawn merely because Frese does not actually state that the server stores the HTML

documents that it provides, then the examiner would almost surely make a rejection under section

103 seeing as storing HTML documents on the same server that serves them was overwhelmingly

well known in the art.

f.   Applicant argues that RDM applet 18 does not generate content because its application

window allegedly cannot reasonably be construed as "contents generated locally." (See Brief at pp.

11).

This argument is not deemed persuasive. The specification does not define the term

"content." The examiner therefore interprets the claimed "content" to mean any information such

as text, data, or graphics displayed by the RDM applet. The RDM applet's application window itself

can arguably be considered graphics. However, the RDM applet also displays other data that more

clearly falls within the broadest reasonable interpretation of "content." For example, the RDM

applet "interrogate[s] a user for identification and a password prior to providing the user access to

an application program." (Frese at col. 8, ll. 14-16). The inherent interrogation interface clearly falls

within the broadest reasonable interpretation of "content."

g.   In the Final Rejection the examiner cited the Ambler reference to show that it was common

in the art to refer to a web page or components thereof as a user interface (Final Rejection mailed

7/3/2007 at pp. 5).

Applicant argues that the web page in Frese is not a "user interface" as claimed because the

reference relied upon by the examiner (Ambler) does not disclose a web page that can be used to

control applications. (Brief at pp. 12)..

This argument is not deemed persuasive. Ambler does not need to show a web page that

controls applications. Ambler was merely relied upon to show that it was common in the art to

refer to web pages as user interfaces in general.

2.   Claim 4

a.   Applicant argues that Frese does not teach all the elements of claim 4 because Frese does

allegedly not mention display of applications installed on the server through the user interface.

(Brief at pp. 13).

This argument is not deemed persuasive. Frese teaches that the HTML documents (i.e., the

user interface(s)) "describe the application programs available for demonstration." (Frese at col. 7,

ll. 34-35).

3.   Claim 6

a.   Applicant argues that Frese does not teach means for generating a merged local client screen.

(Brief at pp. 13). Applicant further argues that this is the case because Frese allegedly does not

mention running any applications on server 20. (Brief at pp. 14).

This argument is not deemed persuasive.

Applicant notes that, in the specification, a merged local client screen includes: (1) a screen area generated locally on client computer 5; and (2) components generated by the interface management program of server 1 and a display of output from applications running on the server 1, if any. (Brief at pp. 13, 14).

Applicant and the examiner appear to generally agree on the broadest reasonable interpretation of a merged local client screen. The examiner interprets the claimed merged local client screen as requiring: (1) a screen area generated locally on the claimed client computer; and (2) display of components generated elsewhere.

Given this interpretation, Frese does teach a merged local client screen. Frese teaches: (1) a browser that has screen area generated locally on the client; and (2) that the browser displays an applet which displays output from applications generated elsewhere. (See, e.g., Frese at fig. 1).

Furthermore, Frese does teach that server 20 can run the applications. Frese states that "RAS 20 launches application programs 22 in an appropriate operating system environment, whether it is on the computer implementing RAS 20 or one coupled to RAS 20" (emphasis added). (Frese at col. 8, ll. 30-32).

4.   **Claims 7-8**

Applicant argues that Frese does not disclose means for controlling the display of a merged local client screen. (Brief at pp. 14).

This argument is not deemed persuasive. Frese discloses that user system 15 has I/O devices such as a mouse, keyboard, or monitor, each of which can reasonably be considered means for controlling the merged local client screen. (Frese at col. 7, ll. 1-5).

5.   **Claim 9**

a.   Applicant argues that Frese lacks means for updating a merged local client screen because Frese allegedly lacks a merged local client screen. (Brief at pp. 15).

This argument is not deemed persuasive because Frese does teach a merged local client screen for the reasons discussed in the section 3 above titled "Claim 6." The examiner refers applicant and the Board to that section for purposes of brevity.

6.   **Claim 18**

a.   In the Final Rejection the examiner took the position that Frese controls some aspects of the display by specifying parameters for the RDM applet (Final Rejection mailed 7/3/2007 at pp. 13).

Applicant argues that this analysis of Frese is improper because "user system 16 controls the display by executing the RDM 18." (Brief at pp. 16).

This argument is not deemed persuasive. Just because user system 16 controls some aspect of RDM applet 18 does not mean that server 20 does not control any aspect of RDM applet 18.

b.   Applicant argues that Frese does not even disclose that the server specifies RDM applet 18's parameters. (Brief at pp. 16).

This argument is not deemed persuasive. Frese shows exemplary HTML received from the server in col. 7, ll. 46-55. That section clearly shows that the HTML received from the server specifies parameters such as "platform" and "colordepth."

c.   Applicant argues that RDM applet 18 does not generate content because its application window allegedly cannot reasonably be construed as "content." (Brief at pp. 16).

This argument is not deemed persuasive. The specification does not define the term

"content." The examiner therefore interprets the claimed "content" to mean any information such

as text, data, or graphics displayed by the RDM applet. The RDM applet's application window itself

can arguably be considered graphics. However, the RDM applet also displays other data that more

clearly falls within the broadest reasonable interpretation of "content." For example, the RDM

applet "interrogate[s] a user for identification and a password prior to providing the user access to

an application program." (Frese at col. 8, ll. 14-16). The inherent interrogation interface clearly falls

within the broadest reasonable interpretation of "content."

7.   Claim 19

a.   Applicant argues that Frese does not disclose "a computer program that, when run on the

computer, causes the computer to accept a user interface provided by the server for controlling the

locally run applications." (Brief at pp. 17).

This argument is not deemed persuasive. Frese teaches a computer program (browser 30)

that, when run on a computer (user system 16), causes the computer (user system 16) to accept a

user interface (web page) provided by a server (RAS 20) for controlling the locally run applications

(applets 18). (See Frese at fig. 1, col. 6, ll. 39 to col. 8, ll. 50).

b.   Applicant argues that Frese does not disclose "a system which displays a screen area having

contents generated locally on the client computer according to display properties specified by the

server." (Brief at pp. 17). Applicant argues that this is the case because "user system 16 of Frese

controls the display by executing RDM 18." (Brief at pp. 17).

This argument is not deemed persuasive. Just because user system 16 controls some aspect

of RDM applet 18 does not mean that server 20 does not control any aspect of RDM applet 18.

Frese discloses that that applet tags in the web page (i.e., user interface) are used to select an

RDM applet that best corresponds to the client's operating environment. Frese shows such tags in

col. 7, ll. 46-55. And, Frese states that the "operating system environment identification is used to

select a RDM which best corresponds to the operating system environment." (Frese at col. 10, ll. 9-

11). For example, the client could select the "rdm", "rdm_internet", or "rdm_ether" applet shown

in col. 7. Because the server provides the applet tags and their parameters that the client is able to

select, the server ultimately controls the look and feel of the applet that is run by the client.

c.   Applicant argues that the web page disclosed by Frese does not meet the claimed user

interface because the web page disclosed by Frese allegedly does not control locally run applications.

(Brief at pp. 17).

The web page disclosed by Frese contains an RDM applet 18 that controls locally run

applications. For example, Frese states that "RDM 18 ... provides a seamless and transparent

interface between the local resource interface of system 16 and application program 22 [that can run

on RAS 20]." (Frese at col. 6, ll. 61-64, col. 8, ll. 30-32).

d.   Applicant argues that the applet parameters disclosed by Frese do not control the display

parameters of RDM applet 18. (Brief at pp. 18).

Frese discloses that that applet tags in the web page (i.e., user interface) are used to select

an RDM applet that best corresponds to the client's operating environment. Frese shows such tags

in col. 7, ll. 46-55. And, Frese states that the "operating system environment identification is used to

select a RDM which best corresponds to the operating system environment." (Frese at col. 10, ll. 9-11). For example, the client could select the "rdm", "rdm_internet", or "rdm_ether" applet shown in col. 7. Because the server provides the applet tags and their parameters that the client is able to select, the server ultimately controls the look and feel of the applet that is run by the client.

B.   U.S. Patent No. 5,613,090 to Willems

The examiner will first summarize the prosecution history as it related to the Willems reference.

In the pre-appeal conference request filed on 18 January 2007 applicant argued the Willems teaches away from modifying the <u>invention of figure 9</u> by making the window manager (100) remote as described by Willems in regards to <u>prior art figure 8</u> because doing so would allegedly contradict the primary stated goal of Willems' invention shown in figure 9, which is allegedly to reduce network traffic.

The examiner agreed with this specific teaching away argument. Willems indeed indicates that the primary stated goal of the <u>invention of figured 9</u> is to reduce network traffic. (Willems at col. 13, ll. 53-58).

However, Willems does <u>not</u> teach away from modifying <u>prior art figure 8</u> to enable the window manager to (1) control a client's locally run applications and (2) run window manager 100 and X server 102 on the same machine.

Regarding modification (1), the examiner's position is that it would have been obvious to enable the window manager to (1) control a client's locally run applications because one of ordinary skill in the art would readily recognize that doing so would reduce the amount of front-end code required. (See Willems at col. 14, ll. 1-5).

Regarding modification (2), the examiner's position is that it would have been obvious to run window manager 100 and X server 102 on the same machine because Willems discloses that having these elements on separate machines results in a performance hit. (See Willems at col. 13, ll. 53-58).

1. Claims 1-10

a. Applicant argues that combining remote window manager 100 and X server 102 on the same remote machine is not consistent with the teachings of Willems. (Brief at pp. 19). Applicant argues that this is the case because "Willems relies on the same reasoning relied on by the Examiner to modify the prior art embodiment of Figure 8 to arrive at the embodiment of Figure 9." (Brief at pp. 19). Applicant argues that the examiner improperly ignores this fact. (Brief at pp. 19).

This argument is not deemed persuasive. The examiner has not ignored Willems disclosure in figure 9. Rather, it is the examiner's opinion that the mere fact that the motivation of reducing front-end code lead Willems to disclose the invention of figure 9 does not establish that one of ordinary skill in the art would consider the invention of figure 9 to be the only way of achieving this result. It does not even establish that Willems himself did not consider locating the window manager 100 and X server 102 on the same machine. Indeed, Willems could very well have considered this modification and found it to be so obvious as to not warrant disclosure.

b. Applicant argues that modifying the prior art embodiment of figure 8 by combining remote window manager 100 and X server 102 on the same machine will increase network traffic contrary to the primary stated goal of Willems. (Brief at pp. 20).

This argument is not deemed persuasive. First, combining remote window manager 100 and X server 102 would clearly <u>decrease</u> network traffic between these two components because there

would no longer be a network separating these components. Second, the examiner reiterates that

the primary stated goal of reducing network traffic is the primary stated goal of the invention of

figure 9 and not of the prior art embodiment of figure 8. Nowhere does Willems teach or fairly

suggest that the primary design goal of prior art figure 8 is to reduce network traffic. To the

contrary, Willems suggests that the fact that the primary purpose of the invention of figure 9 is to

overcome the fact that prior art embodiment of figure 8 does not sufficiently reduce network traffic.

(See Willems at col. 13, ll. 53-58). Moreover, Willems provides a clear motivation to combine

remote window manager 100 and X server 102 where he states that "a performance hit is taken

because of the additional traffic ... between the window manager 100 and the X server 102." (See

Willems at col. 13, ll. 54-58).

c.   Applicant acknowledges the examiner's position that the primary stated goal of reducing

network traffic applies only to the invention of figure 9. (Brief at pp. 21). Applicant takes the

position that this argument does not apply to the present situation because "Willems [allegedly]

provides express motivation not to make the Examiner's proposed change but instead to modify the

Figure 8 embodiment to arrive at the Figure 9 embodiment." (Brief at pp. 21).

This argument is not deemed persuasive. The examiner disagrees with the statement that

"Willems provides express motivation not to make the Examiner's proposed change." In order to

"provide express motivation not to make the Examiner's proposed change" Willems would need to

actually mention the examiner's proposed change. If this were the case then the Willems reference

would anticipate this feature rendering this entire line of arguments moot.

d.   Applicant argues that it would not have been obvious to one of ordinary skill in the art to provide means on the client computer for locally running at least one further application on the basis that this will reduce the amount of front end code required because doing so would allegedly increase network traffic. (Brief at pp. 21).

This argument is not deemed persuasive. The examiner agrees that providing such means would likely increase network traffic. However, again, the primary stated goal of reducing network traffic is the primary stated goal of the <u>invention of figure 9</u> and <u>not</u> of the <u>prior art embodiment of figure 8</u>. And, one of ordinary skill in the art would have readily recognized that the reduction of front-end code is a clear advantage. (See, e.g., Willems at col. 14, ll. 1-5).

e.   Applicant notes that examiner's conclusion that Willems is "configured to enable" the server to control the display on a screen of the display device because Willems' client computer is connected to the server via a network. (Brief at pp. 22). Applicant notes that the same arguments that applied to the Frese reference apply here.

This argument is not deemed persuasive. The same arguments that the examiner made in response to applicant's "configured to enable" arguments in section (A)(1)(a) above apply here. To summarize, claim 1 limit the <u>system</u> (<u>not</u> the server or the client) to being "configured to enable the server ... to control the display ...." For example, claim 1, lines 7-10 recite "the <u>system</u> ... is configured to enable the server (1) to control the display on a screen of the display device (7)..." (emphasis added).

Applicant cites the decision in *Boston Sci. Corp. v. Cordis Corp.* In view of that decision, the examiner acknowledges that in view of this decision "configured to" generally means "intentionally and specifically made to act in a certain way." Here, the <u>system</u> is therefore required to be

intentionally and specifically made to <u>enable</u> the server to control the client's display. Applying the

broadest reasonable interpretation, this means that (1) the server and the client need to be connected

via some network and (2) the server needs to be capable of being provided with means for

controlling the display of the client, as described in the Final Rejection. (See Final Rejection mailed

7/3/2007 at pp. 10). Willems clearly discloses these features and therefore meets the claims. (See,

e.g., Willems at fig. 8).

Moreover, this entire argument is moot because Willems would obviate the claims even if

they actually required the server to control the display on a screen area having contents generated

locally on the client computer because enabling the window manager to control a client's locally run

applications would enable it to control the display on a display on a screen area of the display device

of a screen area having contents generated locally on the client computer. (See Willems at fig. 9, col.

13, ll. 59 to col. 14, ll. 13).

f.   Applicant argues that even if window manager 100 is run remotely it is "not run on X server

102." (Brief at pp. 22). Applicant further argues that "Willems completely lacks a teaching, even in

the prior art embodiment of Figure 8 now relied upon by the Examiner, to configured the server to

provide the user interface." (Brief at pp. 22).

These arguments are not deemed persuasive.

The examiner agrees that, as expressly disclosed by Willems, when window manager 100 is

run remotely it is not run on X server 102. However, combining these components onto the same

machine would clearly have been obvious to one of ordinary skill in the art as the examiner expressly

addressed in the Final Rejection. (See, e.g., Final Rejection mailed 7/3/2007 at pp. 14, second

paragraph under the rejection of claim 1). Again, it would have been obvious to combine remote

window manage 100 and X server 102 to avoid the "performance hit ... taken because of the

additional traffic ... between the window manager 100 and the X server 102." (Willems at col. 13, ll.

54-58).

The examiner disagrees with the statement that "Willems completely lacks a teaching, even

in the prior art embodiment of Figure 8, to configured the server to provide the user interface."

The examiner assumes that by "server" applicant is referring to X server 102. Again, it would have

been obvious to combine remote window manage 100 and X server 102 to avoid the "performance

hit ... taken because of the additional traffic ... between the window manager 100 and the X server

102." (Willems at col. 13, ll. 54-58).

g.   Applicant argues that Willems discloses that the amount of front-end code would be reduced

when control of locally run applications is local and "not when control of locally run applications is

handled by window manager 100" (emphasis added). (Brief at pp. 22-23).

This argument is not deemed persuasive. The examiner disagrees with the statement that

Willems teaches that the amount of front-end code would not be reduced when control of locally

run applications is handled by window manager 100. Willems does not address the result of

enabling a remote window manager to control the locally run applications. One of ordinary skill in

the art is not an automaton. One of ordinary skill would readily recognize that enabling the remote

window manager to control locally run applications would clearly enable the system designer to

eliminate code at the client side.

2.   **Claim 18**

a.   Applicant argues that claim 18 distinguishes over Willems for substantially the same reasons

that applicant argued claim 1 distinguishes over Willems. (Brief at pp. 23-24). Namely, applicant

argues that the skilled person would not modify the embodiment of Figure 8 because doing so

would allegedly "increase network traffic ... directly contradicting the primary stated goal of Willems

to decrease network traffic." (Brief at pp. 23-24).

        The examiner finds these arguments unpersuasive for the same reasons discussed above in

regards to claim 1. To summarize, the primary stated goal of reducing network traffic is the primary

stated goal of the <u>invention of figure 9</u> and <u>not</u> of the <u>prior art embodiment of figure 8, which the</u>

<u>rejection is based upon.</u> <u>Nowhere</u> does Willems teach or fairly suggest that the primary design goal

of <u>prior art figure 8</u> is to reduce network traffic. To the contrary, Willems suggests that the fact that

the primary purpose of the invention of figure 9 is to overcome the fact that prior art embodiment

of figure 8 does <u>not</u> sufficiently reduce network traffic. (See Willems at col. 13, ll. 53-58).

b.   Applicant argues that when window manager 100 is run remotely the server does not control

the display on a screen of the client computer because when window manager 100 is run remotely it

is not run on X server 102. (Brief at pp. 24).

        This argument is not deemed persuasive. The examiner agrees that, as expressly disclosed by

Willems, when window manager 100 is run remotely it is not run on X server 102. However,

combining these components onto the same machine would clearly have been obvious to one of

ordinary skill in the art as the examiner expressly addressed in the Final Rejection. (See, e.g., Final

Rejection mailed 7/3/2007 at pp. 14, second paragraph under the rejection of claim 1). Again, it

would have been obvious to combine remote window manage 100 and X server 102 to avoid the

"performance hit ... taken because of the additional traffic ... between the window manager 100 and the X server 102." (Willems at col. 13, ll. 54-58).

3. Claim 19

a.   Applicant argues that claim 19 distinguishes over Willems for reasons discussed above. The examiner finds these arguments unpersuasive for reasons discussed above. (Brief at pp. 24).

b.   Applicant argues that Willems does not teach that "the computer program, when run on the computer, causes the computer to display a screen area having contents generated locally on the client computer according to display properties specified by the server" because (1) Willems allegedly "does not teach that the windows manager 100 should be located on X server 102", (2) Willems allegedly "does not teach that the display properties of a display should be specified by X server 102", and (3) "controlling the display properties of the display on the local computer using X server 102 would also increase network traffic." (Brief at pp. 24).

These arguments are not deemed persuasive.

As to (1), again, examiner agrees that, as expressly disclosed by Willems, when window manager 100 is run remotely it is not run on X server 102. However, combining these components onto the same machine would clearly have been obvious to one of ordinary skill in the art as the examiner expressly addressed in the Final Rejection. (See, e.g., Final Rejection mailed 7/3/2007 at pp. 14, second paragraph under the rejection of claim 1). Again, it would have been obvious to combine remote window manage 100 and X server 102 to avoid the "performance hit ... taken because of the additional traffic ... between the window manager 100 and the X server 102." (Willems at col. 13, ll. 54-58).

As to (2), window manager 100 controls the display properties. (See Willems at fig. 8, col. 13, ll. 41-44). So, when it is combined with X server 102 as proposed, the server same server controls the display properties.

As to (3), again, the primary stated goal of reducing network traffic is the primary stated goal of the <u>invention of figure 9</u> and <u>not</u> of the <u>prior art embodiment of figure 8, which the rejection is based upon</u>. <u>Nowhere</u> does Willems teach or fairly suggest that the primary design goal of <u>prior art figure 8</u> is to reduce network traffic. To the contrary, Willems suggests that the fact that the primary purpose of the invention of figure 9 is to overcome the fact that prior art embodiment of figure 8 does <u>not</u> sufficiently reduce network traffic. (See Willems at col. 13, ll. 53-58).

c.   Applicant again argues that modifying the prior art embodiment of figure 8 by combining remote window manager 100 and X server 102 on the same machine will increase network traffic, which is allegedly contrary to the primary stated goal of Willems. (Brief at pp. 25).

This argument is not deemed persuasive. First, combining remote window manager 100 and X server 102 would clearly <u>decrease</u> network traffic between these two components because there would no longer be a network separating these components. Second, the examiner reiterates that the primary stated goal of reducing network traffic is the primary stated goal of the <u>invention of figure 9</u> and <u>not</u> of the <u>prior art embodiment of figure 8</u>. <u>Nowhere</u> does Willems teach or fairly suggest that the primary design goal of <u>prior art figure 8</u> is to reduce network traffic. To the contrary, Willems suggests that the fact that the primary purpose of the invention of figure 9 is to overcome the fact that prior art embodiment of figure 8 does <u>not</u> sufficiently reduce network traffic. (See Willems at col. 13, ll. 53-58). Moreover, Willems provides a clear motivation to combine remote window manager 100 and X server 102 where he states that "a performance hit is taken

because of the additional traffic ... between the window manager 100 and the X server 102." (See

Willems at col. 13, ll. 54-58).

d.   Applicant again argues that it would not have been obvious to one of ordinary skill in the art

to provide means on the client computer for locally running at least one further application on the

basis that this will reduce the amount of front end code required because doing so would allegedly ·

increase network traffic. (See Brief at pp. 25).

This argument is not deemed persuasive. Again, the examiner agrees that providing such

means would likely increase network traffic. However, again, the primary stated goal of reducing

network traffic is the primary stated goal of the invention of figure 9 and not of the prior art

embodiment of figure 8. And, one of ordinary skill in the art would have readily recognized that the

reduction of front-end code is a clear advantage. (See, e.g., Willems at col. 14, ll. 1-5).

e.   Applicant argues that it would not have been obvious to provide the client with means for

running local applications because doing so would increase costs and network traffic. (Brief at pp.

25).

This argument is not deemed persuasive. One of ordinary skill in the art is not an

automaton and would readily appreciate that there would be advantages to providing the client

computer with means for running local applications. For example, doing so would enable users to

access applications that may not be available at the server. As to increasing network traffic, again,

the primary stated goal of reducing network traffic is the primary stated goal of the invention of

figure 9 and not of the prior art embodiment of figure 8. And, one of ordinary skill in the art would

have readily recognized that the resulting reduction of front-end code is a clear advantage. (See, e.g.,

Willems at col. 14, ll. 1-5).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related

Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Philip S. Scuderi/
Philip S. Scuderi
Patent Examiner
Technology Center 2100

Conferees:

/NATHAN FLYNN
SUPERVISORY PATENT EXAMINER

Glenton B. Burgess
Supervisory Patent Examiner
Technology Center 2100

For G. Burgess